

## REMARKS

The present invention is directed at how to deal with temporally variant data, and how to get reports that use values for attributes that were believed to be true at a certain point in the past (or that will be true/believed to be true at a point in the future).

This problem has existed for many years. Indeed, US 6 754 657 (Lomet) discusses in columns 1 to 5, and half of column 6, traditional approaches to how to deal with this. By using complex code/software which has layers of temporal integrity rules. Lomet is not really directed at relational databases - because they have layers of relational referential integrity rules as well relating parent and child nodes, for example. Timestamping the start and the end of a value to be used for an attribute is known. And it is the case that the timestamps are altered in known systems - including Lomet if the value is to exist over an altered/changed range. See the Eidelweiss paper referred to later, and in the Information Disclosure Statement..

In Lomet, the Examiner correctly identifies the fact that it does not disclose or teach the feature of claim 3 - inserts used to update and delete timestamps, rather than truly altering the timestamps for start and end values that are already present. In Lomet, the timestamps are set by the system at the time of making a request for information/making a query (requiring complex integrity rules), whereas in the present invention the timestamp for the start time is set at the time of writing the value of the attribute to the database, or at the time of altering the value - at the time of writing an altered value - not at the time of querying the database. In the present invention the timestamps already exist at the time of querying - it is a matter of selecting the correct transaction time, which in turn chooses the timestamps that apply by selecting the correct summary position for that transaction time.

The Examiner refers to US 2003/0208490 (Larrea) to allegedly show inserting new entries for start and end timestamps for a value of an attribute instead of having an actual delete or update operator. With respect, I cannot see any such disclosure in Larrea. When I read paragraph 100 of Larrea referred to by the Examiner, I see:-

"[0100] "Reference Data" is data used that is not expected to change during day-to-day activities, and is most often defined as part of a catalog of data of functionally

related information. Examples of reference data are industry defined data-sets of pharmaceutical information, diagnostic code numbers, sets of government issued identity numbers for persons and physical locations, etc. Reference data is typically used to interpret fields within operating data--thus it is referenced by operating data. While a body of reference data, called a catalog, can be created in the mental health information system, and new entries added over time, such as when a new pharmaceutical comes into use, individual reference data items may not be physically overwritten or deleted according to the present disclosure. A pharmaceutical withdrawn from use, for example, is not physically deleted from the mental health information system, though it may be suppressed from current treatment planning. Were that not the case, then the interpretation of operating data that had used such pharmaceutical data in the past, and that references it, for example, would fail or change post-facto, which is unacceptable in a clinical system."

This is silent on how it technically achieves the ability to access past data. It says it must be capable of viewing the data as it was - but that is what Lomet is all about as well. Lots of people want to be able to view data as it was , and keep the old data. They do not keep the old timestamps unchanged - they alter them to the current values/state for timestamps and keep a log of how to turn them back AT THE TIME OF A QUERY, so that the query views the world with re-created timestamps. They keep an audit trail of changes.

Since Larrea is silent on how it will "turn the clock back ", and since Lomet is not - it will alter the timestamps to currently desired values and turn them back/reinstate /recreate the timestamps that are needed for a historical query at the time of the query, the obvious combination of Larrea and Lomet is to do the same as Lomet (and everyone else). It is not obvious to keep the timestamps for start and end times of a value unchanged and have transaction time - dependent summary position - so that by selecting the transaction time to be used for a query (now, last Wednesday, or November 01, 2004) one selects the timestamps (stored unchanged in memory ) that will be used for all attributes. That is the present invention.

Larrea is about altering data /attribute values , or not altering data/attribute values , and not about whether timestamps are altered over time or stored and a new transaction time with

new start/end timestamps used/created . Larrea's drive for traceability in a medical report is met by the conventional way of handling audit trails of timestamps, as used/disclosed in Lomet. There is no motivation to do anything new. In particular there is no motivation to create additional summary positions for each change and never really delete or update start and end times for attributes. It is using hindsight to see any need to do that in Larrea – he just has a throw away desire not to lose the ability to see what was the clinical recommendation in the past - which the prior art gives him well enough , without having to invent my invention.

In relation to claim 5 , US 2002/0107831 (Sturm) is not concerned with the same thing at all as my invention. It not about how to maintain access to past values of attributes. It does mention null values for values. I have now deleted claim 5 and so its relevance or not is now moot.

In relation to claim 6 , I note that the Examiner, correctly, says that relational databases are know. However, I disagree that it is obvious to use Lomet for a relational database. The software kernel of Lomet to track the temporal integrity issues caused by setting/altering start and end values at the time of analysing the database during a query are bad enough. Adding in the complexity of referential rules will multiply the difficulties many times over - the processing/kernel gets very burdensome.

Indeed, the well-respected author CJ Date feels that certain things just cannot be done in referential/relational databases because of the complexity. Moreover, Lomet does genuinely update and delete his start and end times for values of attributes - which is why he has difficulties keeping of track of how to turn them back. I do not, which is why I have a smaller processing /operation-performing burden on the processor.

My system can do all of the things that CJ Date says cannot be done. That is not obvious. That is surprising and inventive. The realisation that, despite the burden on memory , it is good to keep stat and end timestamps unchanged has freed up my search queries greatly. I now just select the right transaction time for the query. This is simple and elegant.

In relation to claims 8 and 9, US 5 179 650 (Fukui) does not disclose the same overall combination, nor is the claimed combination obvious in view of it: it is silent on timestamps.

Claim 12 has been amended to refer to the fact that for a specific, one attribute, there are a plurality of start times and end times for a plurality of values of the attribute, the different start and end times for the different values existing in the database at the same time and being associated with different transaction (Y) times. This distinguishes from the systems where there may be, for a single attribute, different values at the same time in the database, but only one start and end time in memory - for the current value - historic values/start and end times being accessed by changing /operating on the current start and end times using an audit-trail type program. they do not truly have several start times for the same attribute value field, nor several end times for the same value attribute field.

New claim 21 replaces old claim 1 and has been written with the Examiner's comments in mind. It is a new method of updating or deleting start and end times associated with attributes. The real product of the process is a changed/updated database. Which is capable of being queried so as to display the result of a query to a user. Logical delete or update of start/end timestamps for values is achieved by an insert operation only process, the previous timestamps remaining unaltered. This is novel over the prior art, as acknowledged by the Examiner. I believe that it is not obvious to do this with the timestamps and that the Examiner has read paragraph 100 of Lomet and seen a desire not to lose the previous VALUE of an attribute ( the previous reports/advice) , and imposed my new way of achieving that onto Larrea - which is silent on how it achieves it. Larrea does not mention start time for a value or end time for a value. Larrea would achieve the result to be achieved using the prior art techniques discussed in Lomet.

I enclose, in the Information Disclosure Statement, a copy of documents cited in the corresponding PCT patent application for my invention. In particular, I refer to Edelweiss et al " A Temporal Database Management System Implemented on Top of a Conventional Database, Proceedings of the 20th International Conference of the Chilean Computer Science Society, Santiago, Chile". This addresses the same general sort of background problems as does my invention - as does Lomet, and the prior art discussed in Lomet. Edelweiss changes pre-existing timestamps - see page 61, right hand column, lines 3 - 5,

and lines 20-30, as marked on that citation. Edelweiss has a kernel which does have a genuine update operator. It is not a pure insert only model. Edelweiss points away from the present invention, since it directs the reader to alter the timestamps - along with the other prior art. In relation to the objection to the drawings, the Examiner has correctly identified the fact that the Z axis of Figure 1 is not a time axis. It represents different attributes - it is a schematic way of showing that the X and Y axes apply for each attribute of an entity. The description of Figure 1 on page 38 has been amended to clarify this. Taken with the existing text on page 54, foot of the page, to page 55, especially the third paragraph on page 55 which discusses the Z axes coming out of the page and the fact that the layers of the time cube going into the page represent different attributes of an entity, make the drawings clear. I submit that there is no benefit in drawing a specific Z axis on Figure one - it would only confuse things - the text is clear enough, especially with the amendment now made.

Thus, my invention includes a novel and inventive description of 6NF (called Temporal Normal Form in the invention) implemented using Summary Positions (see Figure 7g). The fact that this invention attempts to define a further decomposition of data is not unique, however the precise definition of how this invention achieves it i.e. the Time Cube or Summary Position is new and inventive in several respects. Consequently the mechanisms required to maintain referential integrity, non-redundancy and non-duplication of operational data in the database are greatly simplified. The technique of adding and maintaining (by altering) timestamps is also known in prior art, however the nature of these mechanisms are complex. In prior art, as logical updates and deletes occur, the timestamps of existing records in the database are physically updated to take account of the new valid start and end times. The very act of physically updating timestamps is a fundamental flaw, which makes the correct management of the operational data in the database very complex. Prior art variously adopts several techniques for the maintenance (changing) of these timestamps to attempt to ensure non-redundancy, non-duplication and referential integrity of the operational data in the database. In contrast, this invention does not require any of these complex mechanisms as it never updates the timestamp values, instead it builds a new Summary Position based on the old Summary Position in a purely insert-only approach (including timestamp values).

Additionally, in prior art, the very fact that data i.e. the timestamps are changed means that the underlying kernel architecture of the database must allow for physical updates of the



timestamps. Therefore all prior art can never adopt a purely insert-only architecture for the Kernel because even though the operational data is only ever inserted (for logical updates and deletes), the associated timestamps are physically updated.

This invention is a purely insert-only approach, including the values of timestamps. This makes the invention novel and gives the invention many advantages over prior art including: -

- The potential to adopt a purely insert-only architecture in the database Kernel so that mechanisms to deal with physical updates are not required.
- Retrieval of data at some point in past or future valid time and/or past transaction times is greatly simplified.
- The mechanisms for management of timestamps to preserve referential integrity, non-redundancy and non-duplication are greatly simplified.

In prior art the concepts of timestamps, valid (x) time, transaction (y) time and insert-only have all been used in various combinations. This invention makes use of this existing knowledge in a unique way and combines it with the novel and inventive technique of a purely insert-only model for management of the timestamps. As a consequence, this invention is much simpler than all previous attempts to maintain referential integrity, non-redundancy and non-duplication of operational data in a database.

As the Examiner will realise, I am only a private individual, not a big company, and if I have any of the technical formatting rules wrong I ask for understanding and forgiveness. I am quite happy to be as helpful as I can - and if given an opportunity to put right procedure will be happy to do so. I hope that the Examiner will also be amenable to finding a way forward. I am sure that my invention is different, and that the present claims delineate the difference and are distinct from the citations. If it would help, I can discuss the invention and the prior art on the telephone.

### **CONCLUSION**

I request a one-month extension of time for replying to the outstanding Official Action and enclose a Credit Card Payment Form authorizing the USPTO to take the fee for the

extension of time, and the fee for submitting an Information Disclosure Statement, from my Credit Card.

Respectfully submitted,

A handwritten signature in black ink, appearing to read 'L. Porter', written over a horizontal dashed line.

Luke M.L. Porter  
46 Margaret Grove  
Harborne  
Birmingham B17 9JL  
Great Britain.

Dated: 14 July 2006